# Minimum cost VRP with time-dependent speed data and congestion charge

**Liang Wen***, **Richard Eglese**

Department of Management Science, Lancaster University Management School, Lancaster LA1 4YX, UK

## Abstract

A heuristic algorithm, called LANCOST, is introduced for vehicle routing and scheduling problems to minimize the total travel cost, where the total travel cost includes fuel cost, driver cost and congestion charge. The fuel cost required is influenced by the speed. The speed for a vehicle to travel along any road in the network varies according to the time of travel. The variation in speed is caused by congestion which is greatest during morning and evening rush hours. If a vehicle enters the congestion charge zone at any time, a fixed charge is applied. A benchmark dataset is designed to test the algorithm. The algorithm is also used to schedule a fleet of delivery vehicles operating in the London area.

**Keywords:** Vehicle Routing; Heuristic; Time-Dependent Data; Congestion Charge

## 1 Introduction

The routing and scheduling of a fleet of vehicles to deliver goods to customers plays an important part in the distribution business. Vehicle operators are concerned with how to minimize the total costs for distributing goods to their customers in order to maximize the operational profit. Many available models assume that the travel times are constant throughout the day, based on constant average speeds. However, this assumption is a weak approximation of the real-world condition where travel time and travel speed are subject to significant variation over time. These variations may result from foreseeable events (e.g., congestion during peak time which increases the fuel cost) or from unforeseeable events like accidents, vehicle breakdowns, transport workers strikes and others. Therefore, the optimal solution to a formulation of the problem that assumes constant travel times may be suboptimal or even infeasible for the time-dependent problem.

*Corresponding author:

E-mail address: l.wen@lancaster.ac.uk (Liang Wen); Telephone: ++44 1524 592758

Congestion charge schemes are applied in various places. For example, London is one of the places applying a congestion charge zone scheme to reduce the traffic in its central area. A congestion charge is one type of road toll that attempts to reduce road congestion. Time-dependent minimum cost vehicle routing problems (VRP) have seldom been addressed in the literature. Including the congestion charge as another factor that impacts the total cost makes the VRP problem even harder to solve. The paper introduces a heuristic method to solve the minimum cost VRP with time-dependent travel times in a road network with a congestion charge zone. The heuristic method is an enhanced version of the LANTIME algorithm which was introduced by Maden et al. [1]. The new heuristic method is called LANCOST.

The rest of the paper is organized as follows: in the next section there is a literature review of relevant previous work about using heuristic methods to solve time-dependent vehicle routing problems. Section 3 describes the LANCOST algorithm to solve a VRP with a time-dependent road network and congestion charge. A benchmark dataset is designed to test the new algorithm. The subsequent section presents computational results obtained through LANCOST using data including real-life traffic information in London. The last section presents conclusions and directions for further research.

## 2 Literature review

The VRP is an NP-hard problem because it includes the travelling salesman problem (TSP) as a special case when only one vehicle with infinite capacity is used to deliver goods. Some VRPs can be solved by exact methods providing they are not too large. However, in practice, the VRP is considerably more difficult to solve than a TSP of the same size. For example, Applegate et al. [2] mention that a TSP involving hundreds and even thousands of vertices can be solved by means of advanced Branch and Cut and Price algorithms. However, the exact algorithm of Baldacci et al. [3] for the VRP can only solve problems with up to about 100 customers, and with a variable success rate. Classical heuristics and metaheuristic methods are used to search for good solutions.

The book of Toth and Vigo [4] provides a comprehensive review of the VRP. It discusses both exact and heuristic methods for the VRP and describes solution approaches for some of the most important variants of the VRP such as the VRP with time windows, backhauls, and pickup and delivery. The book also covers issues arising in real-world VRP applications and includes both case studies and references to software packages. However the book does not

include time-dependent models, so the remainder of the literature review will mostly focus on models where the time to travel between customers is not constant but depends on the time of day. The final section introduces the Vehicle Routing Problem with Multiple Trips which is relevant to the experiments reported in this paper.

Ichoua et al. [5] introduce a time-dependent model for the vehicle routing problem with time windows based on time-dependent travel speeds which satisfies the First In First Out (FIFO) property. They implement a parallel tabu search approach and test its performance both in dynamic and static environments. The scheduling horizon is divided into three time intervals by taking into account the rush hours and three types of road are considered. The results show that the time-dependent model provides significant improvements compared to the model with fixed travel times. Van Woensel et al. [6] modelled VRPs with time-dependent travel times by using a queuing approach. Traffic volumes instead of vehicle speeds are required in the model. The approach is able to deal with a large number of time slots, which improves solution quality.

Eglese et al. [7] show how the use of time-dependent data can affect results for a hypothetical distribution operation and develop a model to use the historical data to construct a Road Timetable that shows the shortest time between nodes when the journeys start at different times. The shortest times and routes may vary as the speed of travel on individual roads may differ significantly by the hour of the day, by the day of the week and by the season of the year. The times required to travel from one end of a road to the other are calculated to ensure that the FIFO property holds and then Dijkstra's algorithm is adapted to find the shortest-time paths between nodes efficiently. The paper describes a case study using real speed data on a road network in the north of England. Maden et al. [1] introduce a vehicle routing and scheduling algorithm, called LANTIME, that is able to accept data from a Road Timetable and to construct a set of vehicle routes that aims to minimize the total time required to deliver goods from a depot to a set of customers subject to a set of constraints. The LANTIME algorithm uses the parallel insertion algorithm firstly to generate an initial feasible solution. Then a tabu search algorithm is used to find better solutions.

Figliozzi [8] has proposed an iterative route construction and improvement algorithm to solve a time dependent vehicle routing problem. The problem is formulated with both soft and hard time windows. The primary objective is to minimize the number of routes and the secondary objective is to minimize total cost, which is proportional to the travel time or travel distance. The solution method is tested on Solomon [9] benchmark problems, but with time-varying

speeds applying to all of the arcs. The algorithm efficiently solves time-dependent vehicle routing problems with time windows and provides a good solution quality when dealing with hard time windows. To further quantify the impacts of congestion on time-dependent real-world networks, this method is applied to a real road network in Conrad and Figliozzi [10]. The results showed that the congestion influences the number of vehicles and travel distances dramatically, especially for depots located outside the customer service area. In order to meet the hard time window constraints, extra vehicles as well as increased travel distances are required. Figliozzi [11] has introduced an emissions vehicle routing problem (EVRP), where departure time and travel speeds are decision variables. The algorithm for solving these problems is to minimize the number of vehicles firstly and then to optimize emissions subject to a fleet size constraint by using a heuristic method. The algorithm is tested using the Solomon [9] benchmark dataset. The results show that significant emissions savings can be achieved by the algorithm. Furthermore, the results indicate that congestion effects on emission levels are not uniform.

Bektas and Laporte [12] do not deal with time-dependent travel times. However they address VRPs with different objective functions that are relevant to this study. They compare four different models with different objectives including distance, energy, weight load and cost minimizing objectives. They provide numerical analyses on some small instances and conclude that minimizing the energy consumption is not equivalent to minimizing the cost. As the labour cost constitutes a major proportion of the total cost, the cost minimizing model focuses on the labour cost in order to reduce the total costs. Advances in engine technology lower the amount and cost of emissions hence lowering the overall total cost. Minimizing the cumulative weight load only does not necessarily imply energy minimization, particularly when time window restrictions are applied.

Estimating greenhouse gas emissions plays an important part in freight transportation planning. Demir et al. [13] review and compare several available freight transportation vehicle emission models and discuss how emissions depend on speed.

The Vehicle Routing Problem with Multiple Trips(VRPMT) is an extension of the classical Vehicle Routing Problem in which each vehicle may perform several routes in the same planning period.The VRPMT was introduced by Fleischmann [14], who used a savings based algorithm to construct the routes and a bin packing heuristic to combine them into working shifts. Followed by using a bin packing heuristic for the working shift aggregation, Taillard et al. [15] introduce a tabu search heuristic to solve the VRPMT which is shown to produce

high quality solutions on a series of test problems. An adaptive memory approach has been proposed by Olivera and Viera [16] where the adaptive memory is a large pool of routes. At each iteration, a subset of these routes is chosen and improved through a tabu search procedure that also aggregates them. The resulting routes are returned into the adaptive memory and the process is iterated. Battarra et al. [17] proposed a adaptive guidance mechanism to reduce the overall number of required vehicles within a limited computing time. Nguyen et al. [18] introduce a tabu search meta-heuristic for the Time-dependent Multi-zone Multi-trip Vehicle Routing Problem with Time Windows. A diversification strategy, guided by an elite solution set and a frequency-based memory, is used to drive the search to potentially unexplored good regions. Azi et al. [19] propose an adaptive large neighborhood search, exploiting the ruin-and-recreate principle, to solve the VRPMT. The algorithm is tested on the well-known benchmark instances proposed by Solomon [9] and other examples to demonstrate the benefits of this multi-level approach.

# 3 Model formulation and algorithm design

## 3.1 Model formulation

The problem considered is a conventional single-depot Vehicle Routing Problem with Time Windows in a time dependent road network. The route for each vehicle starts and ends at the depot. A homogenous fleet of vehicles is used to serve the customers.

The full detail of the model formulation can be found in the PhD thesis of Maden [20]. The main points are summarized below.

The set of customers is denoted by N, the set of vehicles by K and $\{0\}$ represents the depot. For each customer $i \in N$, the service time requirement is s(i), the demand required is w(i) and a time window [e(i),l(i)] is specified for beginning of service. The set of vehicles K represents a homogeneous fleet of vehicles. Each vehicle has capacity W, a starting time $\tau$ and a maximum working time D.

The travelling times between locations are all known and depend on the time when the vehicle starts to travel between the locations. This can be defined as $t_h(i, j)$, where $\{i, j\} \subset N \cup \{0\}$, where h indicates the starting time. The cost $c_h(i, j)$ associated with traversing an arc includes fuel cost, congestion charge and driver cost and replaces travelling time in the original formulation in Maden et al. [1]. The formulation has been extended to

allow the time to travel between locations $t_h(i, j)$ and the travel cost between locations $c_h(i, j)$ to be varied according to the time that the journey is started ($h$).

For each individual vehicle, k, $R_k = [v_0^k, ..., v_{m^k}^k]$ represents the sequence of locations to be serviced by the vehicle k where $v_p^k \in N$ for p=1,…, $m^k - 1$, and, $v_0^k$ and $v_{m^k}^k$ are identified with 0, because the route starts and ends at the depot. $m^k$ represents the number of stops on the complete path, including depot stops, for vehicle k.

The starting time of the service for customer p on the path of vehicle k is denoted by $a(v_p^k)$ and is defined in equation (1). In this equation, h represents the time when the vehicle leaves the previous stop, which is the start time plus the service time at the previous stop, so $h = a(v_{p-1}^k) + s(v_{p-1}^k)$. The service start time for service at stop p is defined as the maximum between (1) the departure time at previous stop, which is $a(v_{p-1}^k) + s(v_{p-1}^k)$ ,plus the travelling time $t_h(v_{p-1}^k, v_p^k)$ to stop p and (2) the start of the time window $e(v_p^k)$ at stop p. The waiting time before a service is denoted as $b(v_p^k)$ and is defined in equation (2). By convention $a(0) = \tau$ and $s(0) = 0$. The waiting time is dependent on both the starting time and the time window start time.

$$a(v_p^k) = \max \begin{Bmatrix} e(v_p^k) \\ a(v_{p-1}^k) + s(v_{p-1}^k) + t_h(v_{p-1}^k, v_p^k) \end{Bmatrix} \quad (1)$$

$$b(v_p^k) = \max \begin{Bmatrix} e(v_p^k) - a(v_p^k) \\ 0 \end{Bmatrix} \quad (2)$$

The only constraint applied directly to each individual customer is that the service must start within the given time window. This is a time window on beginning of service and not on completion of service.

$$e(v_p^k) \le a(v_p^k) \le l(v_p^k), \forall i \in N \quad (3)$$

The working time required on each route must be under the total work time available, D. in the VRPTW the working time includes travelling time, service and waiting time.

$$\sum_{p=0}^{m^k-1} t_h(v_p^k, v_{p+1}^k) + \sum_{p=1}^{m^k-1} s(v_p^k) + \sum_{p=1}^{m^k-1} b(v_p^k) \le D \ \forall k \in K \quad (4)$$

Formula (4) ensures that the vehicle returns to the depot on time. Equation 5 defines the variable $\tilde{R}_k$ as the sequence of locations to be serviced by vehicle k with the depot stops removed.

$$\tilde{R}_k = R_k \setminus \{v_0^k, v_{m^k}^k\} \quad \forall k \in K \tag{5}$$

$$\bigcup_{k \in K} \tilde{R}_k = N \tag{6}$$

The equality constraint (6) ensures that all customers N are dealt with by the individual customer sets $\tilde{R}$.

$$\sum_{p=1}^{m^k-1} w(v_p^k) \leq W \quad \forall k \in K \tag{7}$$

Formula (7) ensures the capacity of the vehicle is not exceeded by the demand requirements of the individual customer sets $R_k$.

The total cost of travelling between locations i and j, starting at time h, is defined as $c_h(i, j)$ where $\{i, j\} \subset N \cup \{0\}$.

The travel cost includes three parts: the fuel cost, driver cost and congestion charge. The fuel cost $F_h(i, j)$ denotes the fuel cost on arc (*i*,*j*) departing at time *h*. $L_h(i, j)$ is the driver cost on arc (*i*,*j*) starting at time h and it is proportional to the time of travel. *CC* is the congestion charge and $\lambda$ is a binary scalar. As the congestion charge is paid once per day, $\lambda = 1$ if it is the first time that a vehicle enters the congestion charge zone, otherwise $\lambda = 0$.Then the travel cost of a vehicle on an $arc(i, j)$ is calculated as follows:

$$c_h(i, j) = F_h(i, j) + L_h(i, j) + \lambda CC \tag{8}$$

The objective function (9) is to minimise the total travelling cost over all the routes.

$$\min \sum_{k \in K} \sum_{p=0}^{m^k-1} c_h(v_p^k, v_{p+1}^k) \tag{9}$$

Compared with the formulation in Maden et al. [1], the objective of the model has been changed from minimizing the total travelling time to minimizing the total cost.

## 3.2 Algorithm design

### 3.2.1 LANTIME Algorithm

The LANTIME algorithm (Maden et al. [1]) solves the least time VRP problem with multiple trips in a road network with time-dependent speeds. A new algorithm, called LANCOST, is based on the LANTIME algorithm and some new enhancements are added to solve the minimum cost version of the problem. The LANCOST algorithm works on Cost Based Road Timetables which are generated by the heuristic methods introduced in Wen et al. [21]. The following section is a description of the LANTIME algorithm, which is the basis for LANCOST.

In LANTIME, an initial solution is created for the VRPTW using the parallel insertion algorithm of Potvin and Rousseau [22]. The parallel insertion algorithm must ensure that the solution is feasible with respect to all constraints.

The initial solution is then improved using a tabu search algorithm. Four possible neighbourhood operations are used in LANTIME: CROSS Exchange, insertion/removal, one exchange and swap.

According to probabilities assigned in advance, the algorithm randomly selects which neighbourhood to explore at each stage. The tabu list is not fixed but varies as proposed by Gendreau et al. [23]; a move added to the tabu list at iteration t is forbidden until iteration $t+\theta$ where $\theta$ is randomly selected from $[\underline{\theta}, \bar{\theta}]$. This device helps to eliminate the probability of cycling between solutions. A standard aspiration criterion overrides the restriction implied by the tabu list if the move leads to a new best solution. A long-term memory structure is used to diversify the search. The tabu search objective has an additional component to represent the long-term memory cost $M(x_{trial})$ of the proposed move that leads to solution $x_{trial}$

$$\text{cost}_{trial} = M(x_{trial})\, f(x_{trial}) \tag{10}$$

The value of $M(x_{trial})$ is higher when $x_{trial}$ has been an accepted move in previous iterations. Thus multiplying the original VRPTW objective $f(x)$ by the memory cost $M(x)$ produces a function with lower values when $x_{trial}$ has never or rarely been accepted before and so helps to diversify the search. The value of $M(x_{trial})$ is multiplied by the value of

a parameter, $\beta$, which thus controls the strength of the effect of the memory cost. Fuller details are provided in Maden et al. [1].

The tabu search objective tries to locate a solution *x* which leads to the minimum search cost. The search cost includes the original VRPTW objective $f(x)$, the memory cost $M(x)$ and the function $P(x)$ that is a measure of the infeasibility of solution *x*.

$$C_{trial} = M(x_{trial}) f(x_{trial}) + \alpha P(x_{trial}) \tag{11}$$

The parameter $\alpha$ is dynamically adjusted throughout the search. It is initially set at 1, in the same way as Gendreau et al. [23]. Then, every $\xi$ iterations, $\alpha = 2\alpha$ if all previous $\xi$ solutions were infeasible, and $\alpha = 1/2\,\alpha$ if all previous $\xi$ solutions were feasible.

### 3.2.2 Differences between LANTIME and LANCOST

The LANTIME algorithm works on one set of Road Timetables which are described in Eglese et al. [7]. The LANCOST algorithm works on two sets of Cost Based Road Timetables. The first of the Timetables finds the least cost route if the congestion charge must be paid for any entry into the congestion charge zone through a link. The route corresponding to this cost may avoid the congestion charge zone or may travel through the congestion charge zone; it will depend on the amount of the congestion charge compared to the additional travel cost incurred by avoiding the zone. The second timetable ignores the effect of the congestion charge when constructing the timetable. The route corresponding to the cost in the second timetable may or may not pass through the congestion charge zone. The reason for two sets of Cost Based Road Timetables is that the vehicle only incurs the congestion charge once per day. Once the congestion charge has been paid, the effect of the congestion charge does not need to be considered. The program uses these two sets of Cost Based Road Timetables to select the least cost route for each vehicle and hence determines whether the vehicle will enter the congestion charge zone on the day in question and so incurs the charge.

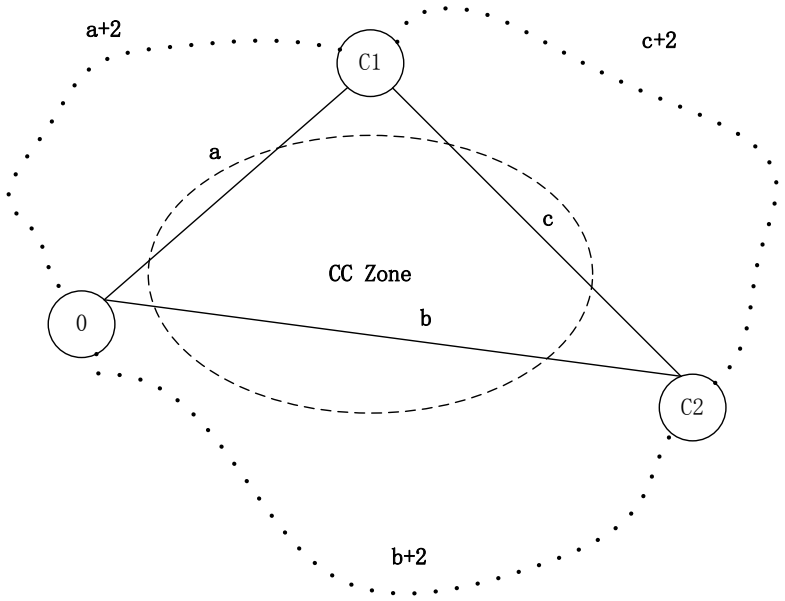


Figure 1 – a small example to demonstrate how the two Cost Based Road Timetables are applied.

In order to have a better idea of the use of these two types of Cost Based Road Timetables, one small example is illustrated in Figure 1. Three nodes and the costs for travelling between each pair of nodes are shown in Figure 1. Node 0 is the depot; the other two nodes (C1 and C2) are customers' nodes. The value of the CC is equal to γ, which is assumed to be greater than 2. The dotted lines are the routes for the Cost Based Road Timetable considering the CC and the minimum costs are described in Table 1. For the purposes of this small illustrative example, costs are the same for travelling between a pair of nodes in each direction. The solid lines are the routes for the Cost Based Road Timetable without considering the CC and the minimum costs are described in Table 2. The vehicle will start from the depot at node 0, and then deliver goods to C1 and C2 and finish at the depot. From Figure 1, the solid lines are going across the CC Zone while the dotted line will drive longer distances to avoid the CC Zone and paying the CC charge. There are two minimum cost routes for both Cost Based Road Timetables. One is 0->C1->C2->0; the other is 0->C2->C1->0. From Table 1 and Table 2, the minimum costs for these routes are a+b+c+6 and a+b+c respectively. But the cost recorded in Table 2 does not include the CC charge. As the corresponding route enters the CC Zone, the CC cost is incurred. So, the total cost is a+b+c+γ. Comparing the results from these two tables, the result from Table 2 will be selected if $\gamma < 6$ and the result from Table 1 will be

selected if $\gamma > 6$. If $\gamma = 6$, then the two routes have equal cost and either route may be selected.

| | 0 | C1 | C2 |
|---|---|---|---|
| 0 | - | a+2 | b+2 |
| C1 | a+2 | - | c+2 |
| C2 | b+2 | c+2 | - |

Table 1. Cost matrix considering CC

| | 0 | C1 | C2 |
|---|---|---|---|
| 0 | - | a | b |
| C1 | a | - | c |
| C2 | b | c | - |

Table 2. Cost matrix without considering CC

The following three additional enhancements are added to the original LANTIME algorithm.

- A method is developed to minimize the number of vehicles going into the congestion charge zone that is described in the next section.
- A complete exploration of the neighbourhood is implemented when the algorithm cannot improve the solution further.
- A 2-opt move algorithm is applied to improve the single routes at the end of the program.

## 3.3 Method for reducing the number of vehicles going into the congestion charge zone

When the congestion charge is considered, the vehicle operators may benefit from fewer vehicles running into the congestion charge zone. However, when developing the solution algorithm, a particular problem encountered was that the heuristic search would not move towards such solutions unless the objective was modified to guide the search in an appropriate direction. So an additional component was added to the objective used by the tabu search. Different methods were tested, but the following version was the one finally used to produce the results.

Let c be the total number of customers from within the congestion charge zone that need to receive a delivery. Let $n_k$ be the number of customers within the congestion charge zone delivered by vehicle k. A negative cost ($-\delta * \sum_{k \in K} \left( \frac{n_k}{c} \right)^2$) is added to the search function. The search function is defined as follows:

$$C_{trial} = M(x_{trial}) f(x_{trial}) + \alpha P(x_{trial}) - \delta \sum_{k \in K} \left( \frac{n_k}{c} \right)^2 \quad (12)$$

$\sum_{k \in K} \left( \frac{n_k}{c} \right)^2$ is less than or equal to 1 and when c is fixed, it will tend to be smaller when there are more non-zero values of $n_i$. Formula (12) is a modification of Formula (11), where the last term $(-\delta \sum_{k \in K} \left( \frac{n_k}{c} \right)^2)$, is added to guide the search so that that less vehicles run into the CC zone. The value of $\delta$ needs to be chosen to relate to the values chosen for the other weighting parameters, $\alpha$ and $\beta$.

## 3.4 Benchmark Datasets

A family of benchmark datasets is constructed where the optimal solution is known from the structure of the datasets. These are used to test whether the LANCOST algorithm can find optimal solutions.

### 3.4.1 4*6 Benchmark Dataset

An initial 4*6 grid based network is constructed and is illustrated in Figure 2. Node 0 is the depot node and each of the other 23 nodes is the location for a customer with demand of one unit. The 38 edges linking the nodes form a rectangular grid and can be traversed in either direction. A congestion charge fee of 5 is imposed for any vehicle entering the central area of the network indicated by dashed lines in Figure 2 and includes edges with one end at nodes 7, 8, 9, 10, 13, 14, 15 and 16. Vehicles must start and finish their routes at the depot (node 0). The cost for a vehicle to travel along any edge is 1. The capacity for each vehicle is 12 units.

The minimum cost for a solution to this bench mark data set is found from the following argument. The total demand is 23 units and the capacity of each vehicle is 12 units, so at least 2 vehicles are required. In order to service a new customer, a vehicle must travel at least one additional unit from the previous customer or depot, so the total distance travelled $\geq 23$. In addition, each of the two vehicles required must return to the depot from the last customer visited, so the total distance travelled $\geq 25$. The structure of the grid means that only 2 customers are within 1 unit of the depot node. So if 2 vehicles must start and finish at the the depot, then 2 of the 4 links from the depot to the first customer or from the last customer to the depot must be at least 2 units, so the total distance travelled $\geq 27$. Also from the structure of the rectangular grid, the length of any vehicle route starting and ending at the depot must

be even. Hence the total distance travelled ≥ 28. At least one vehicle must enter the CC zone to serve the central customers, incurring the CC fee of 5. Hence the total cost ≥28 + 5 = 33.

A feasible solution with total cost of 33 is as follows:

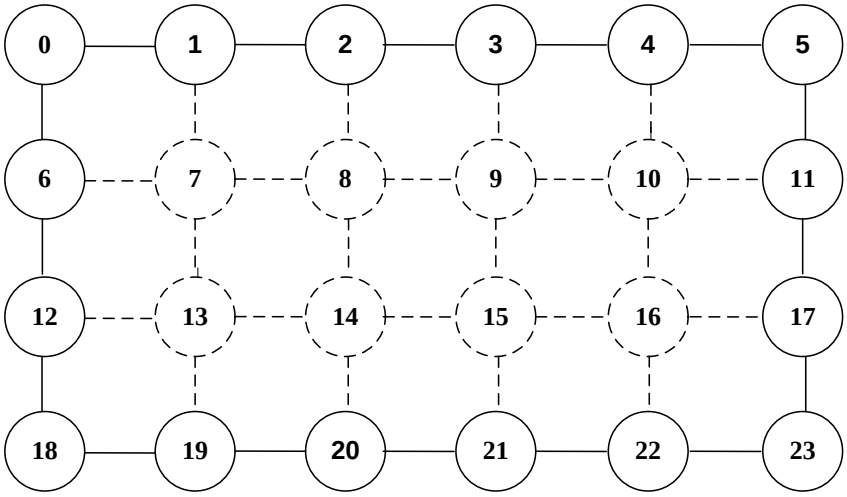


Figure 2 - 4*6 Grid Based Network

Vehicle1 route is 0->1->7->8->9->10->16->15->14->13->12->6->0.

Vehicle2 route is 0->18->19->20->21->22->23->17->11->5->4->3->2->0.

Figure 3 illustrates the solution. The total distance travelled is 12 + 16 = 28 units. Only Vehicle 1 enters the CC zone, so the CC charge is 5 and the total cost of the solution is 28 + 5 = 33. As it has been shown that any feasible solution has a cost of ≥ 33, then this represents an optimal solution.

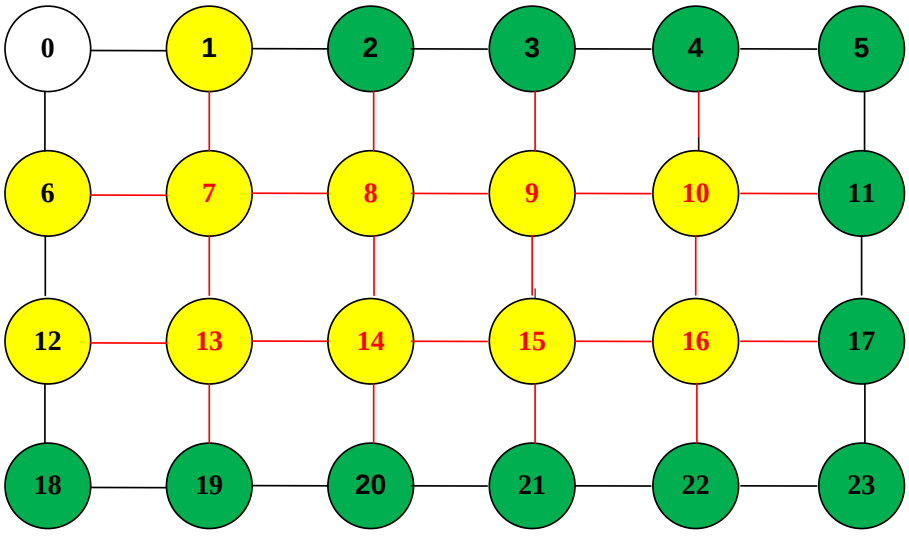


Figure 3 - Solution of 4*6 Benchmark Dataset

The LANCOST algorithm is able to solve the problem and find an optimal solution.

### 3.4.2 Generalised Benchmark Datasets

The initial small example was generalised to test the LANCOST algorithm on larger problems. The row number in the benchmark dataset is fixed to 4. The number of columns can be expanded to N. The total number of nodes is 4N including the depot. So, the total number of customers is 4N -1. Total number of edges is 7N-4. The maximum capacity of a vehicle is $\left\lceil \frac{4N-1}{2} \right\rceil$. CC is the value of the congestion charge. One vehicle will serve all customers within the congestion charge zone, its cost equal to 2N+CC, while the other vehicle serve the rest of the customers at a cost of 2N+4 each, so the total cost is 4(N+1)+CC. This benchmark dataset is designed so that the optimal solution requires one vehicle to deliver goods to all customers within the congestion charge zone.

We can make T copies of the 4*N dataset and each copy is assumed to have a common depot node at 0. The total number of nodes is (4N-1)T+1 including the depot. So, the total number of customers is (4N-1)T. The total number of edges is (7N-4)T. The maximum capacity of each vehicle is $\left\lceil \frac{4N-1}{2} \right\rceil$. There are T*2 vehicles to deliver goods to the customers. T vehicles will serve all customers within the congestion charge zones at a cost of 2N+CC, while another T vehicles serve the rest of the customers costing 2N+4 each, so the total cost is (4(N+1)+CC)T. This benchmark dataset is designed so that the optimal solution requires T vehicles to deliver goods to all customers located in the congestion charge zones.
The method described in Section 3.3 was tested using the dataset with N=20 and T=2. This version of LANCOST was able to minimize the number of vehicles going into the congestion charge zones and solved the benchmark dataset problem with an optimal solution.

# 4 Case Study with Real Traffic Data

## 4.1 Background

The case study is based on the distribution system of a supermarket chain. Customer locations are based on the locations of supermarkets and a distribution centre is located on the outskirts of London.

The type of vehicle used for the case study is a HGV Diesel rigid >32 t EURO 5. The type of vehicle chosen affects the relationship between fuel consumption per kilometre and the speed of the vehicle. Figure 4 shows this relationship for this particular type of vehicle. The formulae used to construct the graph can be found in Department for Transport [24]. The fuel consumption function is convex.

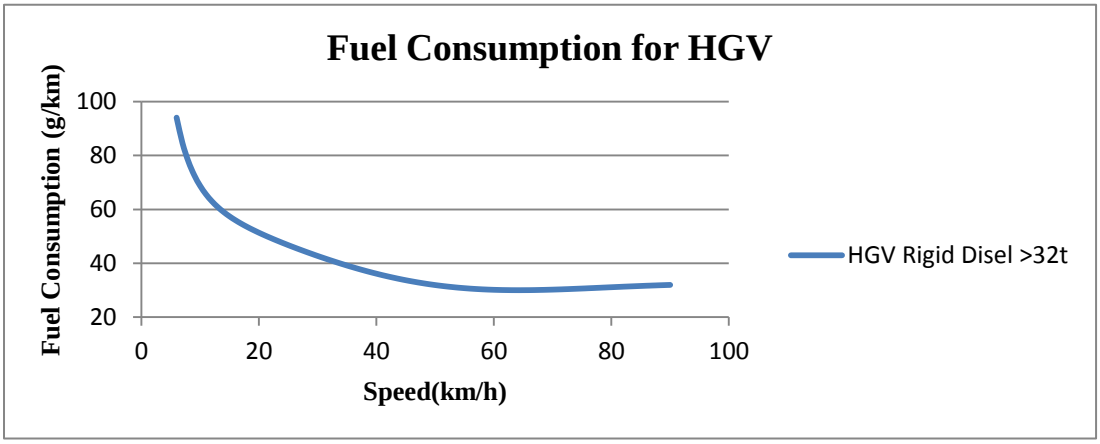


Figure 4 - Fuel Consumption for HGV Diesel Rigid >32 t EURO 5

As the night shift restrictions are applied in the London area, the driver cannot start operations until 7am. Each vehicle can carry a maximum of 24 cages. The unloading time for each cage is 2 minutes. Each driver is available only for a maximum working day of 10 hours including the statutory breaks for driving time and working time.

The objective of the vehicle routing problem is to minimize the total cost. The total cost includes fuel cost, driver cost and congestion charge. There are 10 customers within the congestion charge zone. The vehicle is allowed to wait at the customer locations in order to satisfy the time window constraints. The price of diesel is set to be £1.20 per litre. Fuel consumption is calculated firstly and then converted to CO2e emissions and fuel cost by using the specific conversion factors. From the GHG conversion factor table (2010), one litre of diesel emits 3.1787 kg CO2e. The CO2e can be calculated using the speed along each road in the route using the emissions function provided in Department for Transport [24]. The tables provided allow an estimate to be made of various emission factors in term of emissions per kilometre for different average speeds.

Cost Based Road Timetables were constructed for the set of customers included in each instance tested based on the speeds observed in 96 15-min time bins. The cost based Road

Timetables generate the results for time, distance, cost, CO2e between a pair of locations. The new algorithm works on the cost based Road Timetables to find the minimum cost routes.

Figure 5 shows the customer locations. There are 60 customer locations and 10 of them are in the congestion charge zone. The demands of customers are randomly generated in advance so that the demand for each customer is 4, 5, 6, 7 or 8 cages, each with a probability of 0.2. The customers located in the congestion charge zone have a total demand that can be delivered by only one vehicle, given that each vehicle can perform several trips per day. The distances and time available are such that it is possible for a vehicle to return to the depot to reload after completing all the deliveries for the first trip. The congestion charge is applied from 7am to 6pm during the weekdays. The value of the congestion charge is £8 per vehicle per day. Location 0 is the depot.

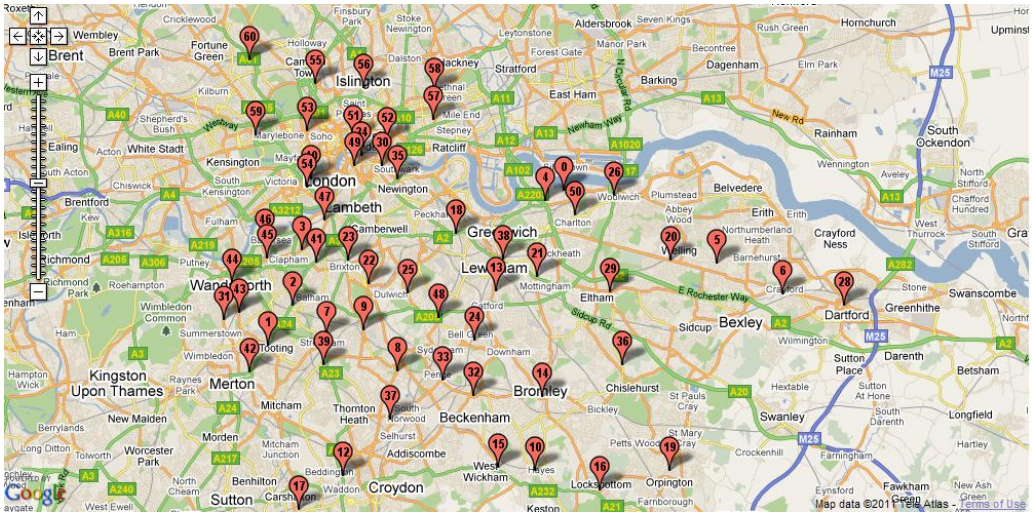


Figure 5 - Customer Locations

The total customer demand is 359 cages. The solution found requires only one vehicle to go into the congestion charge zone. Six vehicles using 16 trips are required to complete the delivery. The routes generated by the new VRP algorithm are shown in Figure 6. The total cost of the fleet is £482.4 with 794kg CO2e emissions. The total distance is 570km. The total driver working time is 2177 minutes. The details of the trips are described in the Appendix A.

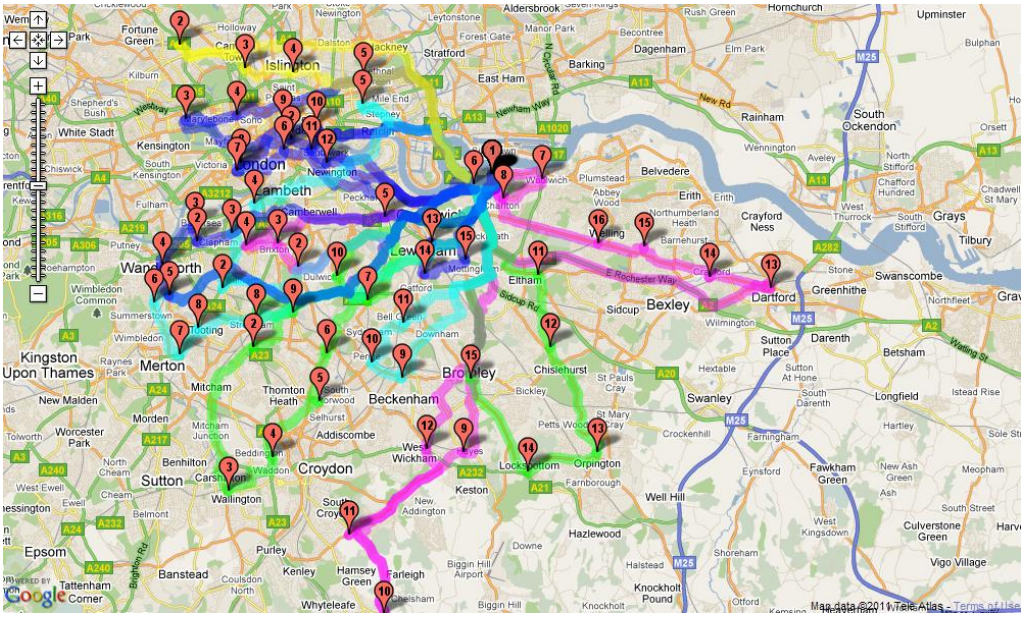


Figure 6 - VRP Routes

## 4.2 Benchmark Dataset Instances

In order to further examine the performance of the LANCOST algorithm, we test the new algorithm by using 10 different benchmark instances. The number of customers served for the 10 benchmark instances ranged from 40 to 45 and were selected from the 60 original customers. The number of vehicles required is up to 10. All of the computations were carried out on a Dual Core processor PC of 3.00GHZ with 3.25GB of RAM and computing times are expressed in seconds. The average computing time for 40 nodes is 4694 seconds and the average computing time for 45 instances is 5566 seconds.

For each instance, three runs were made using the LANCOST algorithm. The first set of runs (A) where the congestion charge is set to 8, the second set of runs (B) where the congestion charge is set to 4, the third set of runs(C) where the congestion charge is set to 0. The traffic conditions are assumed to remain the same even though the value of the congestion charges changes. Changing the value of the congestion charge affects the results of the Cost Based Road Timetable. The results are shown in Table 3. N is the number of customer nodes. As we are using a random number generator, different seeds for the random number generator will give different values of the total cost. The result reported in the table is best one of three runs. As some benchmarks instances do not contain any customers within the congestion charge zone (i.e. Test 3, Test 4, Test 6, Test7), the total congestion charge for Run A is always 0 for these cases. When the value of congestion charge is decreased, the best solution may be for a

vehicle to enter the congestion charge zone in order to save the time and fuel cost with the expense of paying congestion charge, e.g. Test 3 and Test 4 for Run B.

| Data | N | Run | Total time(min) | Total Dist(km) | CO2(kg) | Total cost(£) | Total Congestion charge(£) |
|---|---|---|---|---|---|---|---|
| Test1 | 45 | A | 1677 | 424 | 608 | 374.6 | 8 |
| | | B | 1622 | 426 | 605 | 372.3 | 8 |
| | | C | 1637 | 432 | 610 | 366.2 | 0 |
| Test2 | 45 | A | 1709 | 456 | 640 | 391.9 | 8 |
| | | B | 1608 | 436 | 617 | 377.7 | 8 |
| | | C | 1608 | 436 | 617 | 369.7 | 0 |
| Test3 | 45 | A | 1485 | 418 | 580 | 345.7 | 0 |
| | | B | 1536 | 421 | 589 | 356 | 4 |
| | | C | 1528 | 418 | 582 | 346.2 | 0 |
| Test4 | 45 | A | 1641 | 457 | 634 | 376.4 | 0 |
| | | B | 1546 | 437 | 604 | 362.6 | 4 |
| | | C | 1546 | 437 | 604 | 358.6 | 0 |
| Test5 | 45 | A | 1765 | 448 | 644 | 397.8 | 8 |
| | | B | 1715 | 439 | 634 | 381 | 8 |
| | | C | 1715 | 439 | 634 | 373 | 0 |
| Test6 | 45 | A | 1581 | 438 | 608 | 361.3 | 0 |
| | | B | 1529 | 437 | 600 | 355.6 | 0 |
| | | C | 1529 | 437 | 600 | 355.6 | 0 |
| Test7 | 40 | A | 1400 | 391 | 531 | 313 | 0 |
| | | B | 1336 | 377 | 516 | 304.6 | 0 |
| | | C | 1314 | 374 | 510 | 301.4 | 0 |
| Test8 | 40 | A | 1302 | 343 | 479 | 295.4 | 8 |
| | | B | 1328 | 343 | 485 | 294.9 | 4 |
| | | C | 1280 | 341 | 468 | 278.3 | 0 |
| Test9 | 40 | A | 1525 | 400 | 575 | 354.6 | 8 |
| | | B | 1480 | 397 | 568 | 346.2 | 4 |
| | | C | 1449 | 395 | 561 | 337.2 | 0 |
| Test10 | 40 | A | 1541 | 396 | 571 | 353.4 | 8 |
| | | B | 1457 | 398 | 566 | 344.5 | 4 |
| | | C | 1447 | 397 | 561 | 337.2 | 0 |

Table 3 - Results for 10 different Benchmark Instances

When the congestion charge is set to £8, only one vehicle goes into the congestion zone to deliver goods to customers. When the congestion charge decreases, it is sometimes preferable to allow more vehicles to run into the congestion charge zone. Going through the congestion charge zone saves the distance travelled and hence it saves the fuel costs and time.

| Summary Statistics | | | | | |
|---|---|---|---|---|---|
| Run | Total time(min) | Total dist(km) | Total CO2(kg) | Total cost(£) | Total Congestion charge(£) |
| A | 15626 | 4171 | 5870 | 3564.1 | 48 |
| B | 15157 | 4139 | 5784 | 3495.4 | 44 |
| C | 15053 | 4106 | 5747 | 3423.4 | 0 |

Table 4 - Summary Statistics

Table 4 shows that a higher congestion charge will lead to higher total travelling time, total CO2 emissions and total cost. From the total congestion charge column, Run B does not show much difference from Run A. When the congestion charge price decreases, the best plan

has 11 vehicles (increasing from 6 vehicles) running into the congestion charge zone in order to save cost, travelling distance and reduce CO2 emissions. The total CO2 emission column shows that higher congestion charges lead to higher total CO2 emissions for a local delivery service running from a depot outside a congestion charge zone. The congestion charge increases the CO2 emissions which are an unwelcome consequence of imposing the congestion charge for each vehicle entering the congestion charge zone. These experiments did not consider the effect of changing the level of the congestion charge on traffic patterns and hence speeds. In other words, the traffic conditions are assumed to be the same no matter what level of congestion charge is applied.

In order to test whether the LANCOST algorithm can be used for larger datasets, 10 more experiments have been run. The number of customers in these instances range from 85 to 101 and the locations are selected from actual supermarket locations. The congestion charge is set to £8 and each vehicle can carry up to 24 cages.

The results for these instances are shown in Table 5. The number of customers within the CC zone ranges from 0 to 25. The total number of cages required to be delivered to customers within the CC zone ("CC cages") is shown in Table 5. Dividing this number by the capacity of a vehicle gives a lower bound for the number of vehicle trips into the CC zone. LANCOST is able to reduce the total costs by organizing the trips so that vehicles entering the CC zone often make more than one trip into the CC zone during the day. This strategy reduces the CC paid. The table shows the number of trips for each vehicle that enters the CC zone (though not all trips need enter the CC zone).

| | N | Travel Time | CC customer | CC cages | 1 trip | 2 trips | 3 trips | 4 trips | Travel Distance | CO2 | Travel Cost | CC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Test11 | 85 | 2923 | 0 | 0 | 0 | 0 | 0 | 0 | 853 | 1167 | 693.1 | 0 |
| Test12 | 92 | 3761 | 25 | 146 | 0 | 2 | 4 | 0 | 1070 | 1493 | 939.6 | 32 |
| Test13 | 92 | 3789 | 25 | 142 | 0 | 1 | 3 | 0 | 1087 | 1519 | 939.1 | 32 |
| Test14 | 101 | 3789 | 6 | 38 | 0 | 0 | 2 | 0 | 1112 | 1533 | 926.7 | 16 |
| Test15 | 101 | 3882 | 6 | 35 | 0 | 0 | 2 | 0 | 1130 | 1558 | 941.9 | 16 |
| Test16 | 94 | 3829 | 25 | 150 | 1 | 2 | 2 | 0 | 1111 | 1556 | 971 | 40 |
| Test17 | 90 | 3313 | 5 | 27 | 0 | 0 | 1 | 0 | 950 | 1305 | 784.4 | 8 |
| Test18 | 94 | 3971 | 25 | 150 | 1 | 0 | 4 | 0 | 1129 | 1567 | 974 | 40 |
| Test19 | 94 | 4603 | 25 | 200 | 0 | 2 | 1 | 2 | 1375 | 1931 | 1196 | 40 |
| Test20 | 98 | 3911 | 9 | 62 | 0 | 2 | 2 | 0 | 1116 | 1531 | 941.2 | 32 |
| Total | | 37771 | | | | | | | 10933 | 15160 | 9307 | 256 |

Table 5 - Results for 10 larger instances

## 4.3 Time Windows

Time windows are added to the 10 instance benchmark datasets. A target delivery time is set for each customer. The target delivery time is randomly picked between 8am to 6am and it is always feasible. The goods need to be delivered within the range between target time minus one hour and target time plus one hour. The congestion charge is set to £8. This set of runs is labelled as Run D. The results are shown in Table 6 and Table 7. Compared with Run A, Run D has higher total travelling cost, total distance, travelling time and CO2 emissions.

| Data | N | Run | Total time(min) | Total Dist(km) | CO2(kg) | Total cost(£) | Total Congestion charge(£) |
|---|---|---|---|---|---|---|---|
| Test1 | 45 | A | 1677 | 424 | 608 | 374.6 | 8 |
| | | D | 3431 | 435 | 624 | 392.7 | 16 |
| Test2 | 45 | A | 1709 | 456 | 640 | 391.9 | 8 |
| | | D | 3055 | 471 | 663 | 412.5 | 16 |
| Test3 | 45 | A | 1485 | 418 | 580 | 345.7 | 0 |
| | | D | 3177 | 435 | 608 | 363.7 | 0 |
| Test4 | 45 | A | 1641 | 457 | 634 | 376.4 | 0 |
| | | D | 3076 | 462 | 641 | 389.5 | 8 |
| Test5 | 45 | A | 1765 | 448 | 644 | 397.8 | 8 |
| | | D | 3072 | 465 | 668 | 419.5 | 16 |
| Test6 | 45 | A | 1581 | 438 | 608 | 361.3 | 0 |
| | | D | 3053 | 442 | 615 | 366.8 | 0 |
| Test7 | 40 | A | 1400 | 391 | 531 | 313 | 0 |
| | | D | 2714 | 390 | 531 | 313.1 | 0 |
| Test8 | 40 | A | 1302 | 343 | 479 | 295.4 | 8 |
| | | D | 3273 | 360 | 495 | 310.2 | 16 |
| Test9 | 40 | A | 1525 | 400 | 575 | 354.6 | 8 |
| | | D | 3026 | 415 | 598 | 369.2 | 8 |
| Test10 | 40 | A | 1541 | 396 | 571 | 353.4 | 8 |
| | | D | 2796 | 403 | 577 | 363.9 | 16 |

Table 6- Comparing Run A and Run D

| Summary Statistics | | | | | |
|---|---|---|---|---|---|
| Run | Total time(min) | Total dist(km) | Total CO2(kg) | Total cost(£) | Total Congestion charge(£) |
| A | 15626 | 4171 | 5870 | 3564.1 | 48 |
| D | 30673 | 4278 | 6020 | 3701.1 | 96 |

Table 7- Summary Statistics

Adding time windows to the application makes the problem more complicated. The vehicles spend time waiting at some customer locations in order to satisfy the time windows. The time windows require more than one vehicle to go into the congestion charge zone in some instances.

# 5 Conclusions and Further Research

The new algorithm shows how costs for freight distribution may be significantly influenced by traffic conditions and the presence of a congestion charging scheme. The LANCOST algorithm has been tested on artificially generated networks to check its ability to reduce the number of vehicles entering the congestion charge zone and hence incurring the charge. It has then been used in a case using real traffic data based on a typical supermarket distribution activity in the London area.

The case study has been used to investigate how changing the congestion charge value affects the results. They show that higher values of the congestion charge can lead to more CO2 emissions, more travelling time and more travelling distance for the operator illustrated in this study.

Further research will examine how change to the value of the congestion charge may affect traffic patterns and consider the design of congestion charging schemes.

# Appendix A

The appendix describes the details of trips for each vehicle in the experiment which have been shown in Figure 6. Up to 10 vehicles are available, but only 6 of them are needed.

| Vehicle Name | Arrive | Depart | Location Name | Travel Time | Travel Dist | Travel CO2 | Travel Cost | CC |
|---|---|---|---|---|---|---|---|---|
| **Vehicle 1** | | 07:00 | Depot | | | | | |
| | 07:36 | 07:48 | 39 | 36.87 | 16765 | 23210 | 13.68 | 0 |
| | 08:06 | 08:16 | 17 | 17.36 | 8455 | 11366 | 6.605 | 0 |
| | 08:23 | 08:31 | 12 | 7.285 | 3477 | 4685 | 2.74 | 0 |
| | 08:42 | 08:52 | 37 | 10.94 | 4630 | 6586 | 3.945 | 0 |
| | 09:00 | 09:08 | 8 | 7.764 | 2799 | 4259 | 2.643 | 0 |
| | 09:42 | 09:42 | Depot | 34.27 | 14763 | 21019 | 12.51 | 0 |
| | 10:04 | 10:18 | 48 | 22.16 | 10293 | 14032 | 8.253 | 0 |
| | 10:31 | 10:43 | 7 | 12.95 | 6435 | 8506 | 4.938 | 0 |
| | 10:49 | 11:03 | 9 | 5.983 | 2157 | 3329 | 2.055 | 0 |
| | 11:12 | 11:20 | 25 | 8.891 | 3886 | 5378 | 3.216 | 0 |
| | 11:47 | 11:47 | Depot | 26.98 | 11007 | 15888 | 9.596 | 0 |
| | 12:00 | 12:10 | 29 | 12.79 | 7445 | 9135 | 5.154 | 0 |
| | 12:16 | 12:26 | 36 | 6.652 | 3607 | 4600 | 2.624 | 0 |
| | 12:40 | 12:48 | 19 | 13.35 | 6212 | 8362 | 4.937 | 0 |
| | 12:55 | 13:03 | 16 | 7.463 | 3817 | 4975 | 2.873 | 0 |
| | 13:44 | 13:56 | 14 | 10.53 | 5625 | 7196 | 4.12 | 0 |
| | 14:16 | | Depot | 20.75 | 12027 | 14617 | 8.285 | 0 |
| | | | | | | | | |
| **Vehicle 2** | | 07:00 | Depot | | | | | |
| | 07:27 | 07:35 | 34 | 27.95 | 11979 | 17140 | 18.2 | 8 |
| | 07:52 | 08:04 | 59 | 16.49 | 6870 | 9930 | 5.947 | 0 |
| | 08:13 | 08:27 | 53 | 8.978 | 2823 | 4638 | 2.948 | 0 |
| | 08:50 | 09:02 | 58 | 23.22 | 6934 | 11662 | 7.499 | 0 |
| | 09:27 | 09:27 | Depot | 24.7 | 10352 | 13518 | 8.396 | 0 |
| | 09:59 | 10:09 | 49 | 32.15 | 12988 | 19137 | 11.51 | 0 |
| | 10:18 | 10:34 | 54 | 9.372 | 3324 | 5117 | 3.182 | 0 |
| | 10:38 | 10:48 | 40 | 3.253 | 749 | 1409 | 0.9658 | 0 |
| | 10:57 | 11:09 | 51 | 9.228 | 3221 | 5025 | 3.127 | 0 |
| | 11:42 | 11:42 | Depot | 33.44 | 12675 | 18903 | 11.6 | 0 |
| | 12:10 | 12:24 | 52 | 27.82 | 10872 | 15912 | 9.717 | 0 |
| | 12:30 | 12:46 | 30 | 5.679 | 1244 | 2297 | 1.625 | 0 |
| | 12:49 | 13:05 | 35 | 3.436 | 1096 | 1761 | 1.123 | 0 |
| | 14:03 | 14:03 | Depot | 27.47 | 9658 | 15167 | 9.388 | 0 |
| | 14:15 | 14:29 | 38 | 12.34 | 5487 | 7657 | 4.536 | 0 |
| | 14:49 | 15:05 | 13 | 5.1 | 1836 | 2866 | 1.762 | 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 15:12 | 15:22 | 21 | 7.157 | 2590 | 3977 | 2.456 | 0 |
| | 15:32 | | Depot | 9.742 | 5081 | 6598 | 3.79 | 0 |
| | | | | | | | | |
| **Vehicle 3** | | 07:00 | Depot | | | | | |
| | 07:36 | 07:50 | 2 | 36.85 | 16800 | 23352 | 13.73 | 0 |
| | 07:59 | 08:11 | 3 | 8.192 | 3094 | 4641 | 2.844 | 0 |
| | 08:17 | 08:29 | 47 | 6.099 | 2032 | 3279 | 2.051 | 0 |
| | 09:07 | 09:17 | 57 | 38.08 | 12118 | 19370 | 12.39 | 0 |
| | 09:39 | 09:39 | Depot | 21.86 | 9209 | 12667 | 7.696 | 0 |
| | 10:27 | 10:43 | 31 | 48.46 | 22169 | 30535 | 17.99 | 0 |
| | 10:52 | 11:06 | 42 | 8.942 | 3224 | 4846 | 3.022 | 0 |
| | 11:10 | 11:26 | 1 | 4.314 | 1487 | 2370 | 1.47 | 0 |
| | 12:13 | 12:13 | Depot | 46.33 | 19345 | 27737 | 16.65 | 0 |
| | 12:38 | 12:48 | 32 | 25.15 | 14097 | 17626 | 10.01 | 0 |
| | 12:51 | 13:03 | 33 | 3.595 | 1648 | 2306 | 1.35 | 0 |
| | 13:41 | 13:55 | 24 | 7.179 | 3332 | 4580 | 2.686 | 0 |
| | 14:33 | | Depot | 22.97 | 11793 | 15237 | 8.816 | 0 |
| | | | | | | | | |
| **Vehicle 4** | | 07:00 | Depot | | | | | |
| | 07:25 | 07:35 | 22 | 25.75 | 12199 | 16679 | 9.73 | 0 |
| | 07:40 | 07:52 | 23 | 4.775 | 1813 | 2678 | 1.648 | 0 |
| | 07:57 | 08:07 | 41 | 5.375 | 1794 | 2839 | 1.788 | 0 |
| | 08:32 | 08:42 | 18 | 24.94 | 7600 | 12212 | 7.935 | 0 |
| | 09:01 | 09:01 | Depot | 18.94 | 6916 | 10657 | 6.549 | 0 |
| | 09:04 | 09:20 | 4 | 2.989 | 1327 | 1871 | 1.105 | 0 |
| | 09:29 | 09:39 | 26 | 8.823 | 3984 | 5433 | 3.227 | 0 |
| | 09:43 | 09:59 | 50 | 4.057 | 2348 | 2921 | 1.644 | 0 |
| | 10:04 | 10:04 | Depot | 4.41 | 1506 | 2342 | 1.472 | 0 |
| | 10:35 | 10:47 | 10 | 31.2 | 16311 | 20940 | 12.06 | 0 |
| | 11:04 | 11:16 | 11 | 17 | 11340 | 12783 | 7.093 | 0 |
| | 11:24 | 11:36 | 27 | 8.237 | 5076 | 5947 | 3.344 | 0 |
| | 11:47 | 11:57 | 15 | 10.71 | 6647 | 7810 | 4.377 | 0 |
| | 12:27 | 12:27 | Depot | 30.31 | 17235 | 21186 | 12.04 | 0 |
| | 12:49 | 12:57 | 28 | 21.93 | 18786 | 20450 | 10.64 | 0 |
| | 13:36 | 13:48 | 6 | 9.107 | 3443 | 5117 | 3.146 | 0 |
| | 13:56 | 14:10 | 5 | 8.432 | 4143 | 5472 | 3.19 | 0 |
| | 14:17 | 14:29 | 20 | 6.075 | 2463 | 3538 | 2.146 | 0 |
| | 14:44 | | Depot | 15.39 | 6798 | 9177 | 5.517 | 0 |
| | | | | | | | | |
| **Vehicle 5** | | 07:00 | Depot | | | | | |
| | 07:50 | 08:02 | 60 | 50.99 | 21306 | 29942 | 18.1 | 0 |
| | 08:15 | 08:29 | 55 | 12.43 | 3886 | 6401 | 4.074 | 0 |
| | 08:39 | 08:53 | 56 | 10.13 | 2819 | 4844 | 3.179 | 0 |
| | 09:31 | 09:31 | Depot | 37.9 | 14499 | 20376 | 12.75 | 0 |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Vehicle 6** | 07:00 | 07:00 | Depot | | | | | |
| | 07:42 | 07:52 | 45 | 42.72 | 16415 | 24569 | 14.97 | 0 |
| | 07:55 | 08:09 | 46 | 2.706 | 826 | 1353 | 0.8719 | 0 |
| | 08:18 | 08:32 | 44 | 9.122 | 2871 | 4706 | 2.993 | 0 |
| | 08:37 | 08:47 | 43 | 5.021 | 1528 | 2518 | 1.62 | 0 |
| | 09:39 | | Depot | 51.91 | 21256 | 30814 | 18.55 | 0 |
| | | | | | | | | |
| | | | | 36:17:00 | 570 | 794 | 482.4 | 8 |

Table A.1

# References


[1] Maden W, Eglese R, Black D. Vehicle routing and scheduling with time-varying data: A case study. Journal of the Operational Research Society 2010; 61:515-522.
[2] Applegate D, Bixby R, Chvatal V, Cook W. The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics). Princeton University Press; 2007.
[3] Baldacci R, Christofides N, Mingozzi A. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming 2008; 115:351-385.
[4] Toth P, Vigo D. An overview of vehicle routing problems. The vehicle routing problem 2002; 9:1-26.
[5] Ichoua S, Gendreau M, Potvin JY. Vehicle dispatching with time-dependent travel times. European Journal of Operational Research 2003; 144:379-396.
[6] Van Woensel T, Kerbache L, Peremans H, Vandaele N. Vehicle routing with dynamic travel times: A queueing approach. European Journal of Operational Research 2008; 186:990-1007.
[7] Eglese R, Maden W, Slater A. Road Timetable (TM) to aid vehicle routing and scheduling. Computers & Operations research 2006; 33:3508-3519.
[8] Figliozzi M. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. Transportation Research Part E: Logistics and Transportation Review 2012; 48:616-636.
[9] Solomon MM. Algorithms for the vehicle-routing and scheduling problems with time window constraints. Operations Research 1987; 35:254-265.
[10] Conrad RG, Figliozzi MA. Algorithms to Quantify Impact of Congestion on Time-Dependent Real-World Urban Freight Distribution Networks. Transportation Research Record: Journal of the Transportation Research Board 2010; 2168:104-113.
[11] Figliozzi M. Vehicle routing problem for emissions minimization. Transportation Research Record: Journal of the Transportation Research Board 2010; 2197:1-7.
[12] Bektas T, Laporte G. The pollution-routing problem. Transportation Research Part B: Methodological 2011; 45:1232-1250.
[13] Demir E, Bektaş T, Laporte G. A comparative analysis of several vehicle emission models for road freight transportation. Transportation Research Part D: Transport and Environment 2011; 16:347-357.
[14] Fleischmann B. The vehicle routing problem with multiple use of vehicles. In: Fachbereich Wirtschaftswissenschaften, Universität Hamburg 1990.
[15] Taillard ED, Laporte G, Gendreau M. Vehicle routeing with multiple use of vehicles. Journal of the Operational Research Society 1996:1065-1070.
[16] Olivera A, Viera O. Adaptive memory programming for the vehicle routing problem with multiple trips. Computers & Operations research 2007; 34:28-47.
[17] Battarra M, Monaci M, Vigo D. An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. Computers & Operations research 2009; 36:3041-3050.
[18] Nguyen PK, Crainic TG, Toulouse M. A tabu search for Time-dependent Multi-zone Multi-trip Vehicle Routing Problem with Time Windows. European Journal of Operational Research 2013; 231:43-56.
[19] Azi N, Gendreau M, Potvin J-Y. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Computers & Operations research 2014; 41:167-173.
[20] Maden W. Models and heuristic algorithms for complex routing and scheduling problems. In: Management School. Lancaster University; 2006.
[21] Wen L, Çatay B, Eglese R. Finding a minimum cost path between a pair of nodes in a time-varying road network with a congestion charge. European Journal of Operational Research.

[22] Potvin J-Y, Rousseau J-M. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research 1993; 66:331-340.
[23] Gendreau M, Hertz A, Laporte G. A tabu search heuristic for the vehicle routing problem. Management science 1994; 40:1276-1276-1290.
[24] Department for Transport. 2009. *Road vehicle emission factors:Regulated* [Online]. Available: http://www.dft.gov.uk/publications/road-vehicle-emission-factors-2009/ [Accessed September 15 2013].